

# Large Data Visualization with PAT: Programmable Analysis Tool

Patrick Moran

Data Analysis Group

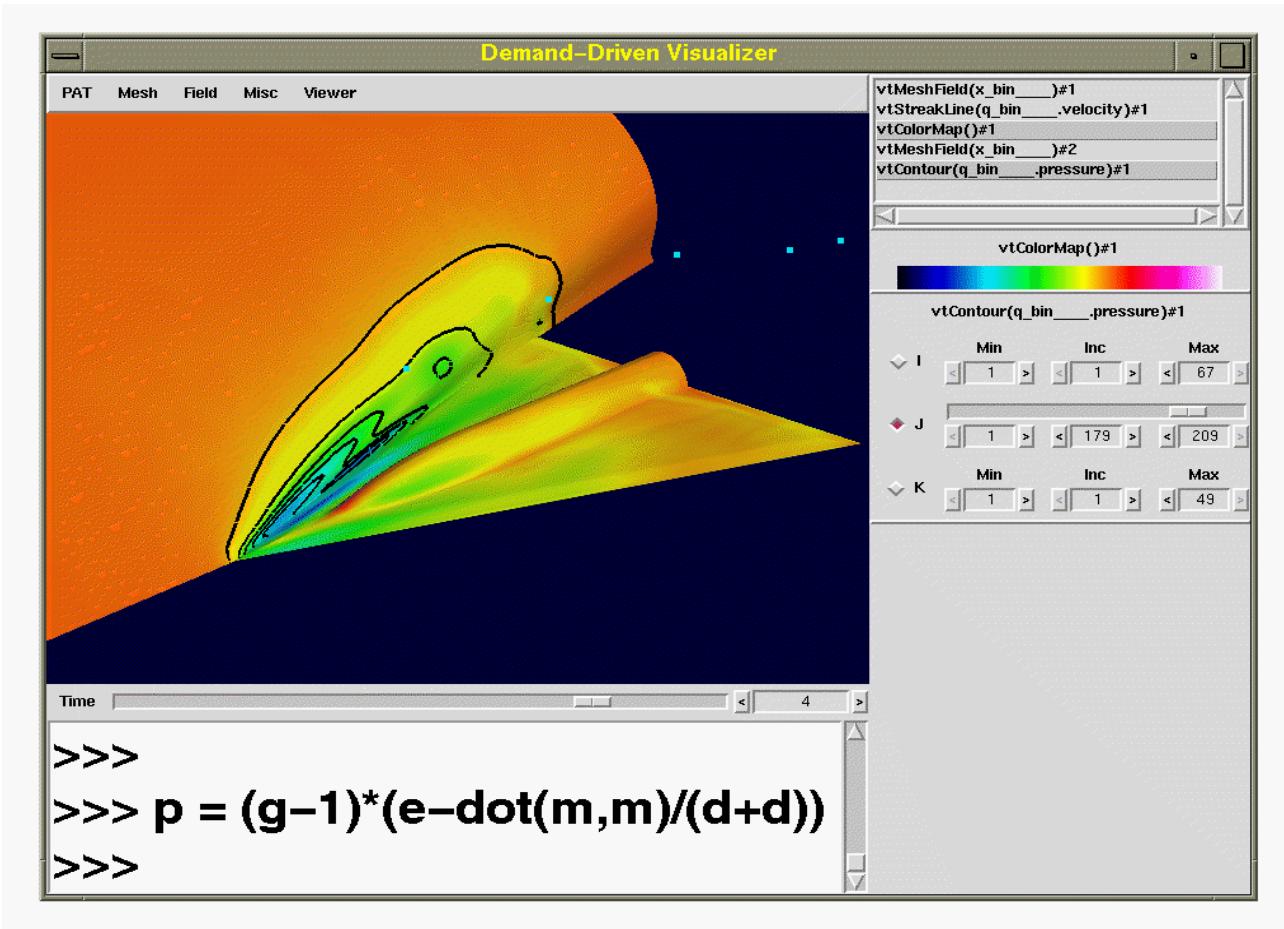
<http://www.nas.nasa.gov/Groups/VisTech>

NASA Ames Research Center

<http://www.nas.nasa.gov/~pmoran>



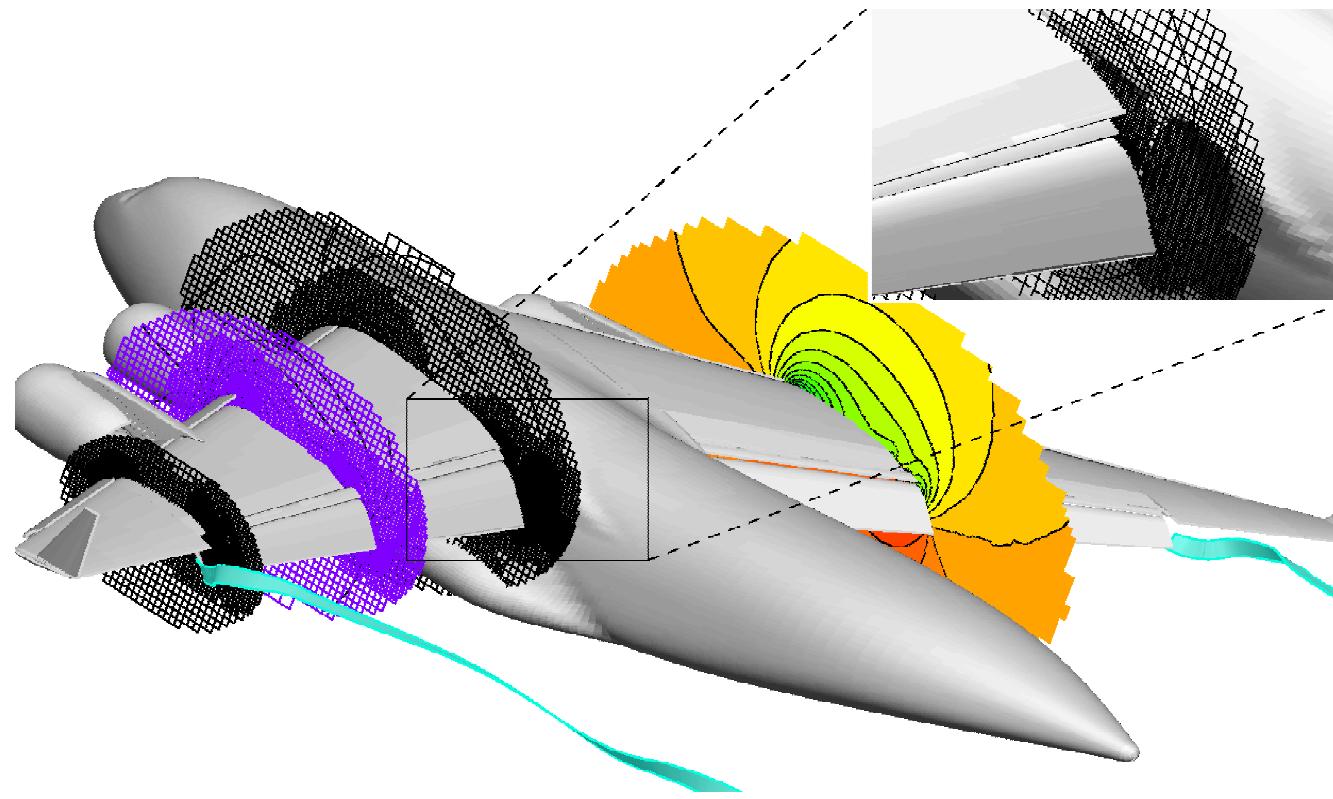
# Big Data



<http://www.nas.nasa.gov/~pmoran>



# Garden of Grids

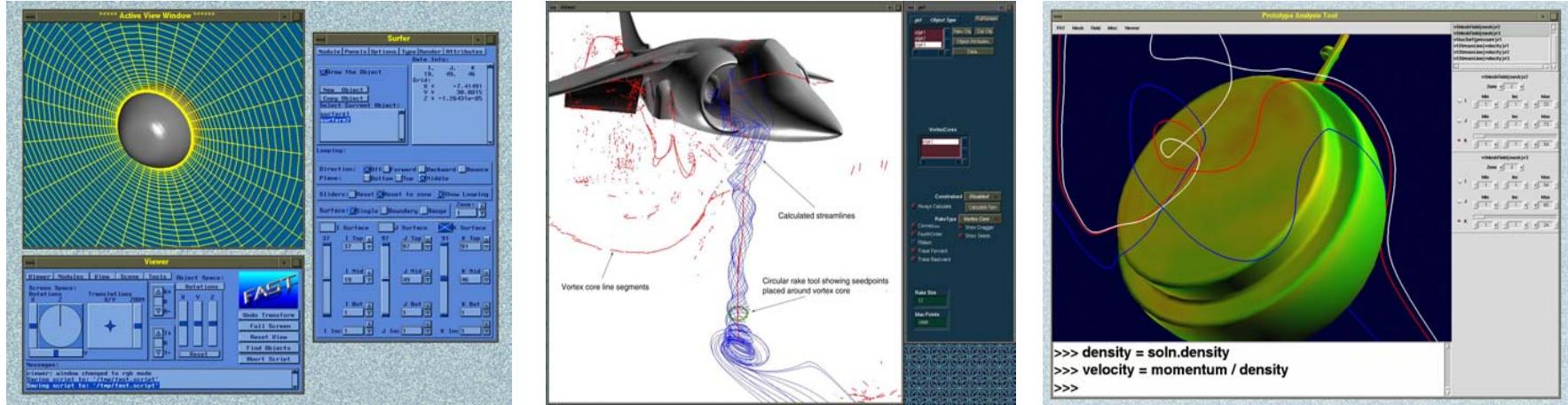


- e.g., CART3D, High-wing transport
  - M. Aftosmis

<http://www.nas.nasa.gov/~pmoran>



# Previous Work

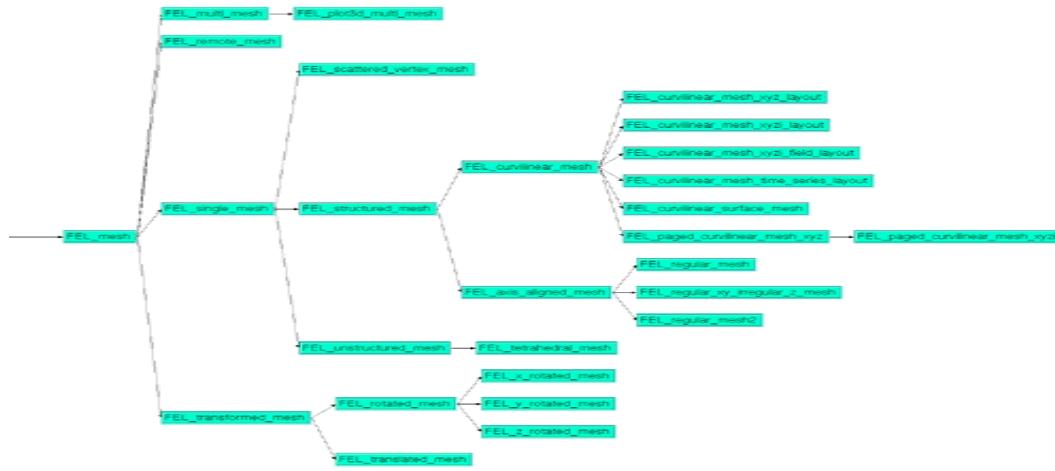


- FAST
- Gel
- *Prototype Analysis Tool*

<http://www.nas.nasa.gov/~pmoran>



# Initial Components



- Field Encapsulation Library (FEL2)
  - VisTech
  - graphics, direct-manipulation libraries



# Learnings

- + C++ and templates
- + out-of-core paging (Ellsworth & Cox)
- + demand-driven philosophy
  
- need more general data model
- PLOT3D assumptions creep
- software distribution issues

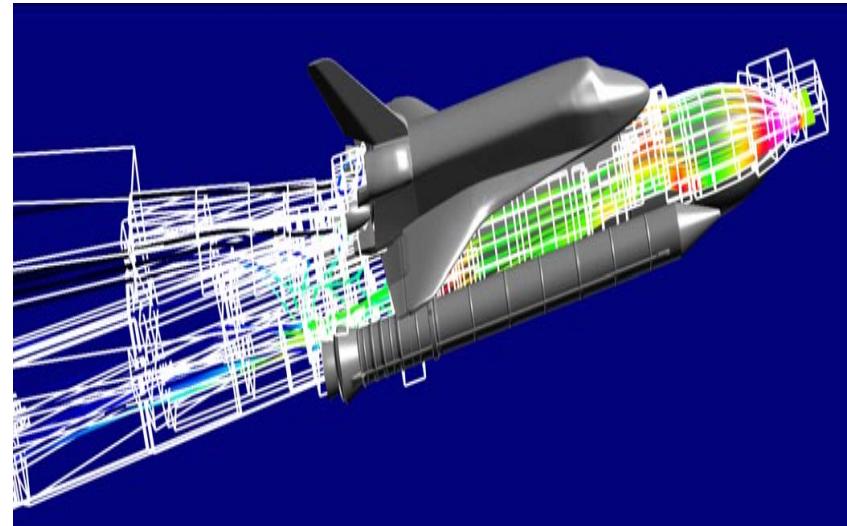
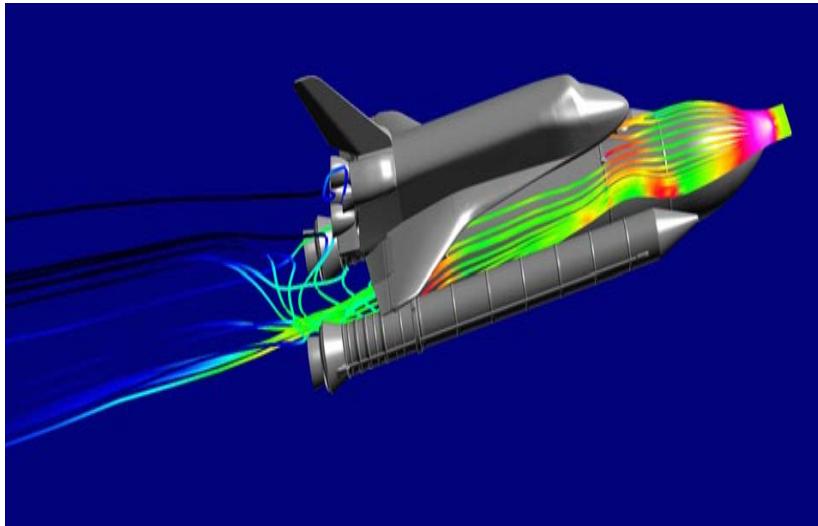


# Current Approach

- C++ components composed in Python
- demand-driven philosophy
- modular design
- flexible field model
- open source



# Demand-Driven Evaluation

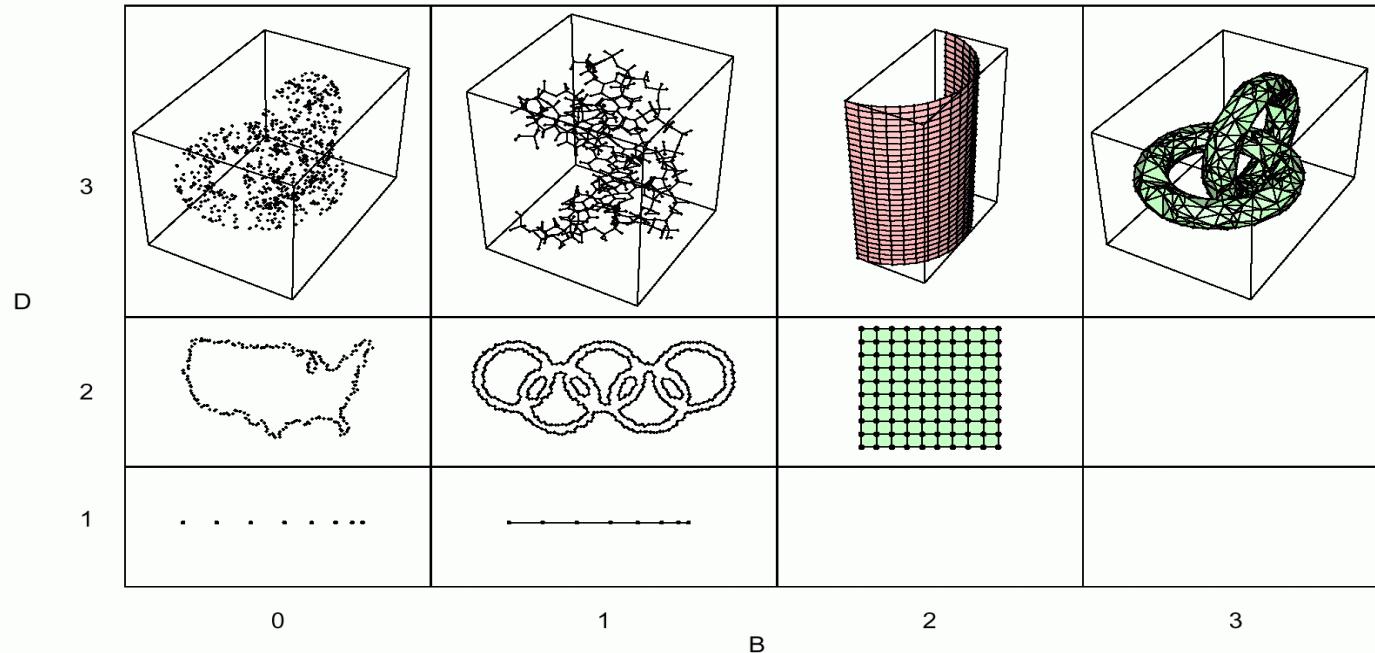


- many techniques touch small % of data
- often interested in derived values
- time-series data accentuate issues

<http://www.nas.nasa.gov/~pmoran>



# Field Model (FM)



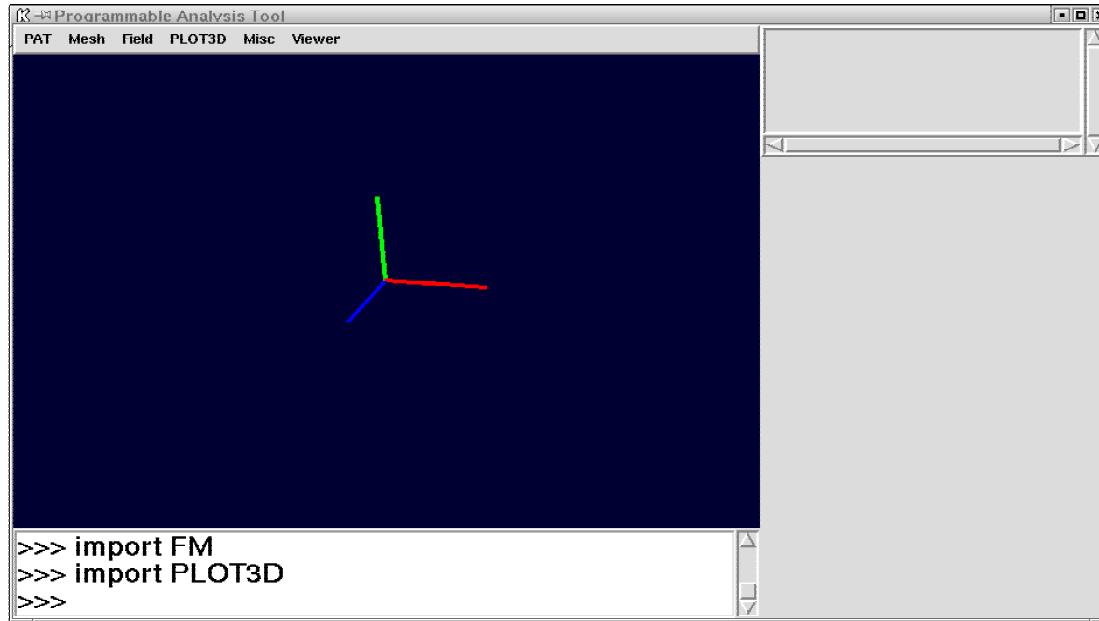
- C++, templated base (B) and phys (D) dim.
- meshes with not necessarily manifold shapes
- work with data in-place

# Python

- object-oriented
- interactive, interpreted
- extensible
- active user community
- many additional modules
- open source



# *Programmable Analysis Tool*



- process capture, replay, programmability
- modularity, dynamic module import
- rapid prototyping

<http://www.nas.nasa.gov/~pmoran>



# Design Challenges in PAT

- direct interpreter window
- maintaining consistent state
- beginner users, advanced users
- parallel execution, serial GUI
- distributed visualization



# SourceForge

- provides CVS repository host
- bug tracking, statistics, releases
- home to several relevant projects:
  - Python
  - PyOpenGL
  - Numeric
  - Chromium (parallel rendering)



# SourceForge Statistics

July 18, 2001

- number of projects: 23,737
- number of registered users: 214,561
- page views: 1,423,131
- files downloaded: 197,929
- fifth most active project: Python



# FM on SourceForge

The screenshot shows a web browser window with the following details:

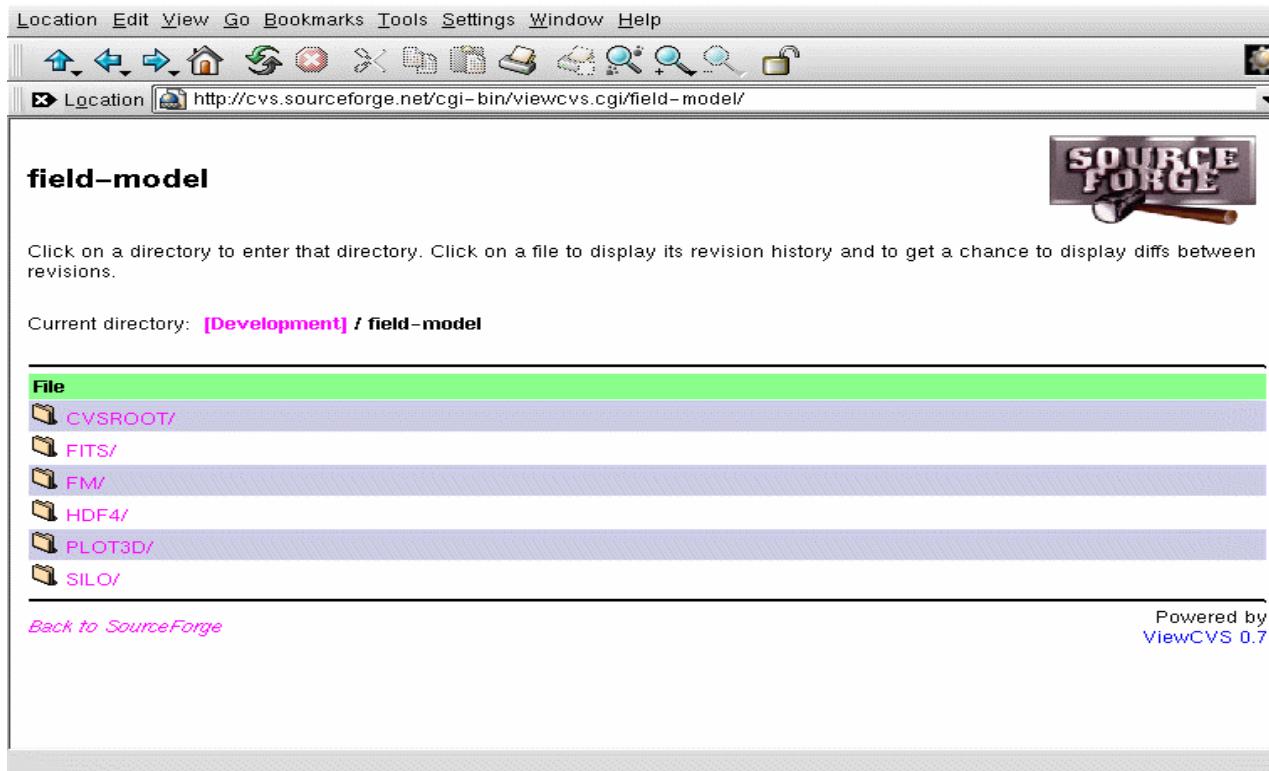
- Location bar:** http://field-model.sourceforge.net
- Toolbar:** Standard browser icons for back, forward, search, and file operations.
- Content Area:**
  - Header:** "Field Model" with two green logo icons on either side.
  - Text:** "Field Model (FM) is a C++ library with classes representing fields and the meshes that fields are based on. FM is a component enabling the interchange of data that come in a variety of mesh and field types. FM features:"
  - List:**
    - a general, extensible model for field data
      - contributors can enhance the library with new object types
    - an object-oriented, modular, templated C++ implementation
      - templating enables new field types to be introduced with greater code reuse
    - interface abstractions applicable to a broad variety of field and mesh types
      - algorithms written in terms of the field abstractions may work with a broad variety of objects
    - a general, STL-compatible iterator model, where meshes behave as collections of cells
      - we leverage libraries now standard to C++ and user familiarity with those libraries
    - a design with many optimizations for large data
      - e.g., a consistent, demand-driven evaluation philosophy
    - support for field data where field values are associated with hexahedra and other non-vertex cells
      - e.g., native support for "cell-centered" data, without the need for dual-mesh techniques
    - multi-threaded capability
      - designed so that large field and mesh objects can be shared reliably and efficiently
  - Text:** "FM is a follow-on project to the [FEL2 project](#). We learned a great deal during that project, now we "throw one away" and start fresh. One primary goal of this project is to develop a foundation set of classes for field data such that others can contribute code built on that foundation. For example, our hope is that others can contribute file I/O code for the file formats specific to their environment. The incentive would be that once one's data is brought into the common field model, then one can leverage shared tools built in terms of the field interfaces."
  - Text:** "See our [Source Forge projects page](#) for more."
  - Bottom Left:** "SOURCE FORGE™" logo.
  - Bottom Center:** "Loading complete"

- <http://field-model.sourceforge.net/>

<http://www.nas.nasa.gov/~pmoran>



# FM Module Organization



- future candidates: CARD3D, Vis5D, CGNS, HDF5, PV3, TAG2D, TAG3D, etc.

<http://www.nas.nasa.gov/~pmoran>



# Plans

- continue to fill out FM classes
- port paging to FM
- port VisTech to FM
- cultivate collaborators
- more documentation

